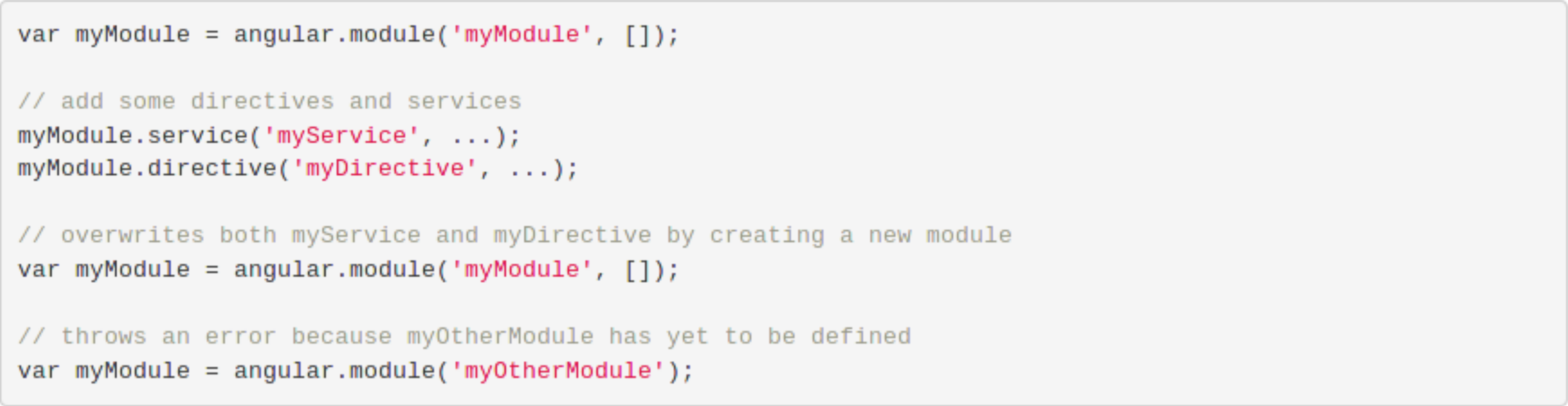
ANGULARJS MEETS PYTHON

ANGULARJS

by Google

HTML enhanced for web apps!

Modules



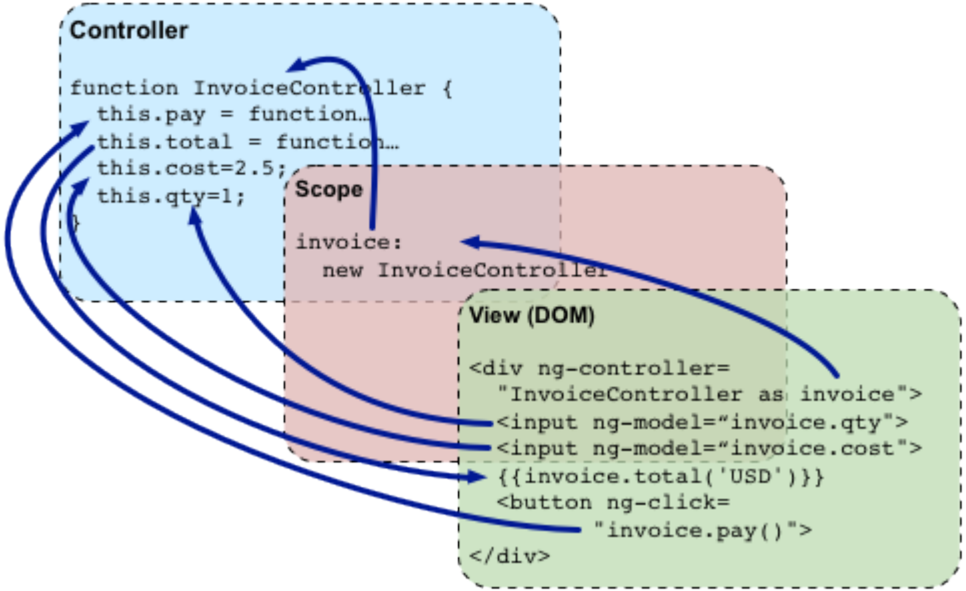
```
var myModule = angular.module('myModule', []);

// add some directives and services
myModule.service('myService', ...);
myModule.directive('myDirective', ...);

// overwrites both myService and myDirective by creating a new module
var myModule = angular.module('myModule', []);

// throws an error because myOtherModule has yet to be defined
var myModule = angular.module('myOtherModule');
```

Controllers

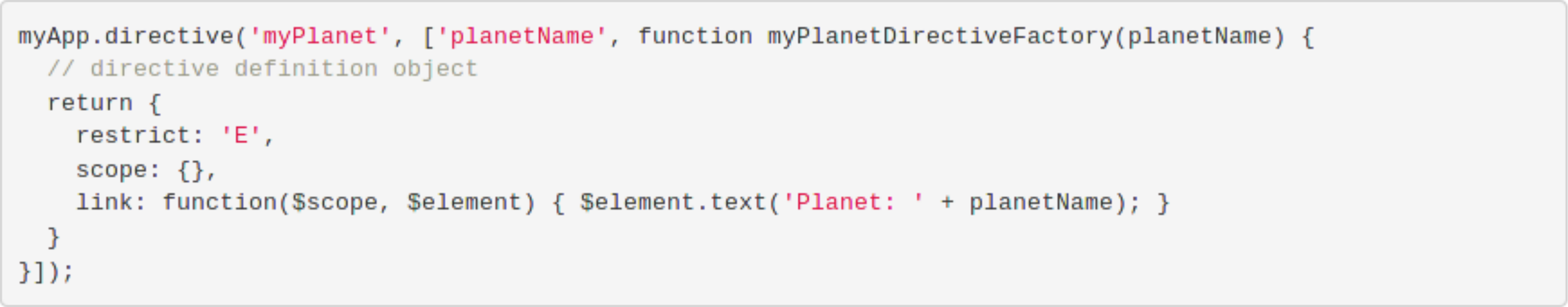


```
function InvoiceController {
  this.pay = function…
  this.total = function…
  this.cost=2.5;
  this.qty=1;
}
```

```
invoice:
  new InvoiceController
```

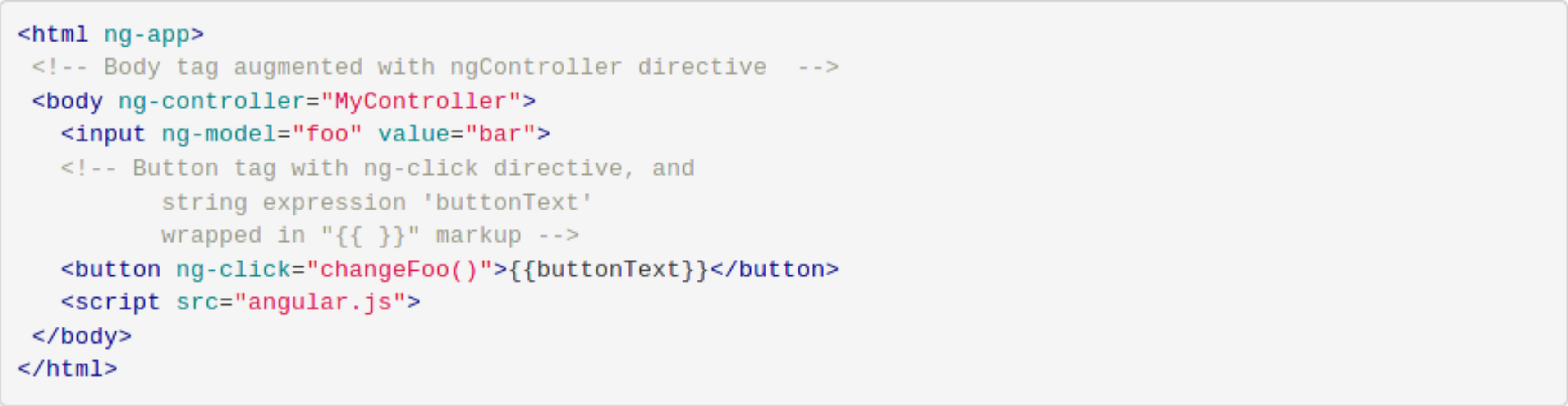
```
<div ng-controller=
  "InvoiceController as invoice">
  <input ng-model=“invoice.qty">
  <input ng-model=“invoice.cost">
  {{invoice.total('USD')}}
  <button ng-click=
       "invoice.pay()">
</div>
```

Directives



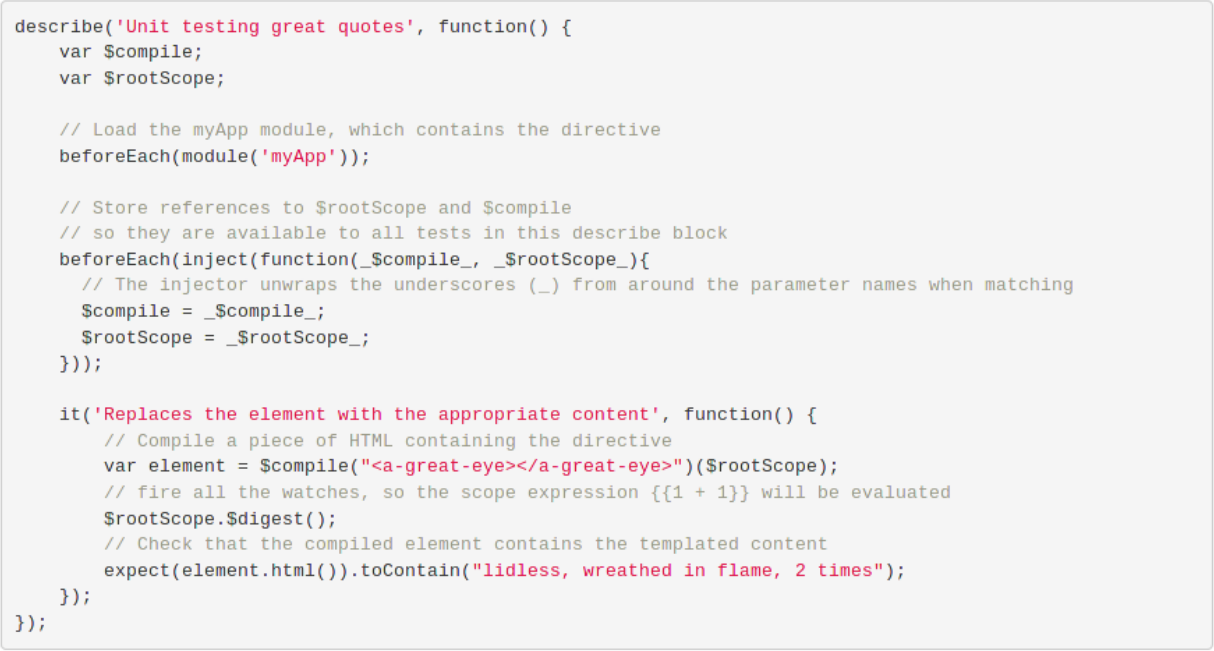
```
myApp.directive('myPlanet', ['planetName', function myPlanetDirectiveFactory(planetName) {
  // directive definition object
  return {
    restrict: 'E',
    scope: {},
    link: function($scope, $element) { $element.text('Planet: ' + planetName); }
  }
}]);
```

Templates



```
<html ng-app>
 <!-- Body tag augmented with ngController directive  -->
 <body ng-controller="MyController">
   <input ng-model="foo" value="bar">
   <!-- Button tag with ng-click directive, and
          string expression 'buttonText'
          wrapped in "{{ }}" markup -->
   <button ng-click="changeFoo()">{{buttonText}}</button>
   <script src="angular.js">
 </body>
</html>
```

Testing



```
describe('Unit testing great quotes', function() {
    var $compile;
    var $rootScope;

    // Load the myApp module, which contains the directive
    beforeEach(module('myApp'));

    // Store references to $rootScope and $compile
    // so they are available to all tests in this describe block
    beforeEach(inject(function(_$compile_, _$rootScope_){
      // The injector unwraps the underscores (_) from around the parameter names when matching
      $compile = _$compile_;
      $rootScope = _$rootScope_;
    }));

    it('Replaces the element with the appropriate content', function() {
        // Compile a piece of HTML containing the directive
        var element = $compile("<a-great-eye></a-great-eye>")($rootScope);
        // fire all the watches, so the scope expression {{1 + 1}} will be evaluated
        $rootScope.$digest();
        // Check that the compiled element contains the templated content
        expect(element.html()).toContain("lidless, wreathed in flame, 2 times");
    });
});
```

GRACIAS

@circun4